# Roots of Two Nonlinear Equations for the HP-41C

To find the roots for the following two simultaneous nonlinear equations:

$f(x,y) = 0$
$g(x,y) = 0$

let $Fx(x,y) = df(x,y)/dx = (f(x+hx,y) - f(x,y)) / hx$
let $Fy(x,y) = df(x,y)/dy = (f(x,y+hy) - f(x,y)) / hy$
let $Gx(x,y) = dg(x,y)/dx = (g(x+hx,y) - g(x,y)) / hx$
let $Gy(x,y) = dg(x,y)/dx = (g(x,y+hy) - g(x,y)) / hy$

Where,
$hx = 0.001 * (1 + |x|)$
$hy = 0.001 * (1 + |y|)$

To refine the guesses for x and y use the following equations:
$J = Fy(x,y) * Gx(x,y)- Fx(x,y) * Gy(x,y)$
$x = x - (Fy(x,y) * g(x,y) - f(x,y) * Gy(x,y)) / J$
$x = y - (Gx(x,y) * f(x,y) - g(x,y) * Fx(x,y)) / J$

## Algorithm

Input: x, y, TolerX, TolerY, and MaxIter

```
Iter = 0
Do
  Iter = Iter + 1
  f = f(x,Y)
  g = g(x,Y)
  h = 0.001 * (1+|y|)
  fx = (f(x+h,y) - f)/h
  gx = (g(x+h,y) - g)/h
  h = 0.001 * (1+|y|)
  fy = (f(x,y+h) - f)/h
  gy = (g(x,y+h) - g)/h
  J = fy * gx - fx * gy
  DiffX = (fy * g - f * gy) / J
  DiffY = (gx * f - g * fx) / J
  x = x - DiffX
  y = y - DiffY
Until (|DiffX| < TolerX and |DiffY| < TolerY) or
      (Iter > MaxIter)
Return Iter, x, y
```

# HP-41C Implementation

## Instructions

Note 1: Numeric input and output is designated by red text.

Note 2: Alphanumeric text is enclosed in double quotes.

| *Step* | *Input* | *Command* | *Output* |
|---|---|---|---|
| 1 | Move program pointer to label F | [GTO][F] | |
| 2 | Switch to program mode. | [PRGM] | |
| 3 | Edit the commands between LBL F and RTN to implement your f(x,y) function. | | |
| 4 | When done switch out of program mode. | [PRGM] | |
| 5 | Move program pointer to label G | [GTO][G] | |
| 6 | Switch to program mode. | [PRGM] | |
| 7 | Edit the commands between LBL G and RTN to implement your g(x,y) function. | | |
| 8 | When done switch out of program mode. | [PRGM] | |
| 9 | Start the program. | XEQ “RT2” | X↑Y? |
| 9b | To rerun the program. | [A] | X↑Y? |
| 10 | Enter initial guess for x. | x[ENTER] | |
| 11 | Enter initial guess for y. | y[R/S] | TLRX↑TLRY? |

| *Step* | *Input* | *Command* | *Output* |
|---|---|---|---|
| 12 | Enter tolerance limit for x. | TolerX[ENTER] | |
| 13 | Enter tolerance limit for y. | TolerY[R/S] | MAX ITRS? |
| 14 | Enter maximum number of iterations. | MaxIter[R/S] | |
| 15 | Program displays the number of iterations.<br>☞ If the iterations exceed the maximum limit, the program plays the audible tone 9 and displays the message **ITER MAX REACHED**. To resume viewing the number of iterations, press [R/S]. | | ITERS=iterations |
| 16 | To view the root for variable x. | [R/S] | X=x |
| 17 | To view the root for variable y. | [R/S] | Y=y |
| 18 | To investigate other possible roots, go to step 9. | | |
| 19 | To solve for a different functions f(x,y) and/or g(x,y) go to step 1. | | |

## Example

Note: Since the example uses the existing code for functions f(x,y) and g(x,y) found I labels F and G, we will start with the step that simply runs the programs. The code calculates the values for the following functions:

f(xy) = x^2 + y^3 – 31 = 0

g(x,y) = x * y – 6 = 0

The example uses the following input:

- Initial guess for x is 5.
- Initial guess for y is 5.
- Tolerance for the root of variable x is 1E-8.
- Tolerance for the root of variable y is 1E-8.
- The maximum number of iterations is 55.

| *Step* | *Comment* | *Command* | *Output* |
|---|---|---|---|
| 1 | Start the program. | XEQ "RT2" | X↑Y? |
| 2 | Enter 5 for the initial guess for x. | 5 [ENTER] | |

| Step | Comment | Command | Output |
|---|---|---|---|
| 3 | Enter 5 for the initial guess for y. | 5 [R/S] | TLRX↑TLRY? |
| 4 | Enter 1E-8 for the tolerance limit for x. | 1[EEX]8[CHS][ENTER] | |
| 5 | Enter 1E-8 for the tolerance limit for y. | 1[EEX]8[CHS][R/S] | MAX ITRS? |
| 6 | Enter 55 for the maximum number of iterations. | 55[R/S] | |
| 7 | The program displays the number of iterations. | | ITERS=7. |
| 8 | To view the root for variable x. | [R/S] | X=2.00000 |
| 9 | To view the root for variable y. | [R/S] | Y=3.00000 |

## Memory Map

```
R00 = x
R01 = y
R02 = hx, hy, J
R03 = f(x,y)
R04 = g(x,y)
R05 = Fx(x,y)
R06 = Fy(x,y)
R07 = Gx(x,y)
R08 = Gy(x,y)
R09 = Iter
R10 = Toler x
R11 = Toler y
R12 = Delta x
R13 = Delta y
R14 = MaxIter
```

## Listing

Note: You my insert PSE or VIEW commands in suitable code locations to view intermediate values for the refined guesses of the roots for x and/or y.

| *Program Step* | *Comment* |
|---|---|
| ♦LBL "RT2" | |
| ♦LBL A | |
| "X↑Y?" | |
| PROMPT | Prompt for initial guesses |
| STO 01 | |
| X<>Y | |
| STO 00 | |
| "TLRX↑TLRY?" | Prompt for tolerance values |
| PROMPT | |
| STO 11 | |
| X<>Y | |
| STO 10 | |
| "MAX ITRS?" | Prompt for the maximum number of iterations |
| PROMPT | |
| STO 14 | |
| 0 | |
| STO 09 | I = 0 |
| ♦LBL 00 | Start the main loop |
| RCL 01 | |
| RCL 00 | |
| XEQ F | Calculate f(x,y) |
| STO 03 | Store f(x,y) |
| RCL 01 | |
| RCL 00 | |
| XEQ G | Calculate g(x,y) |
| STO 04 | Store g(x,y) |
| RCL 00 | |
| ABS | |
| 1 | |
| ST+ 09 | I = I + 1 |
| + | |
| .001 | |
| * | |

| ***Program Step*** | ***Comment*** |
|---|---|
| STO 02 | h = 0.001 * (1 + ABS(x)) |
| RCL 01 | |
| RCL 00 | |
| RCL 02 | |
| + | |
| XEQ F | Calculate f(x+h,y) |
| RCL 03 | |
| - | |
| RCL 02 | |
| / | |
| STO 05 | Store Fx(x,y) |
| RCL 01 | |
| RCL 00 | |
| RCL 02 | |
| + | |
| XEQ G | Calculate g(x+h,y) |
| RCL 04 | |
| - | |
| RCL 02 | |
| / | |
| STO 07 | Store Gx(x,y) |
| RCL 01 | |
| ABS | |
| 1 | |
| + | |
| .001 | |
| * | |
| STO 02 | h = 0.001 * (1 + ABS(y)) |
| RCL 01 | |
| + | |
| RCL 00 | |
| XEQ F | Calculate f(x,y+h) |
| RCL 03 | |
| - | |
| RCL 02 | |
| / | |
| STO 06 | Store Fy(x,y) |
| RCL 01 | |

| ***Program Step*** | ***Comment*** |
|---|---|
| RCL 02 | |
| + | |
| RCL 00 | |
| XEQ G | Calculate g(x,y+h) |
| RCL 04 | |
| - | |
| RCL 02 | |
| / | |
| STO 08 | Store Gy(x,y) |
| RCL05 | |
| * | |
| RCL 06 | |
| RCL 07 | |
| * | |
| X<>Y | |
| - | |
| STO 02 | J = fy * gx - fx * gy |
| 1/X | Put 1/J in the stack |
| RCL 06 | |
| RCL 04 | |
| * | |
| RCL 03 | |
| RCL 08 | |
| * | |
| - | |
| * | Multiply (fy * g - f * gy) by 1/J |
| STO 12 | DiffX = (fy * g - f * gy) / J |
| ST- 00 | X = X - DiffX |
| RCL 07 | |
| RCL 03 | |
| * | |
| RCL 04 | |
| RCL 05 | |
| * | |
| - | |
| RCL 02 | |
| / | |
| STO 13 | DiffY = (gx * f - g * fx) / J |

| *Program Step* | *Comment* |
|---|---|
| `ST- 01` | Y = Y - DiffY |
| `RCL 14` | |
| `RCL 09` | |
| `X>Y?` | Is Iter > MaxIter? |
| `GTO 01` | Exit loop |
| `RCL 10` | |
| `RCL 12` | |
| `ABS` | |
| `X>Y?` | Is ABS(DiffX) > TolerX |
| `GTO 00` | Resume the next iteration |
| `RCL 11` | |
| `RCL 13` | |
| `ABS` | |
| `X>Y?` | Is ABS(DiffY) > TolerY |
| `GTO 00` | Resume the next iteration ---------- end of main loop |
| `♦LBL 02` | Display results |
| `"ITERS="` | |
| `FIX 0` | |
| `ARCL 09` | |
| `FIX 5` | |
| `PROMPT` | Display the number of iterations |
| `"X="` | |
| `ARCL 00` | |
| `PROMPT` | Display root for variable x |
| `"Y="` | |
| `ARCL 01` | |
| `PROMPT` | Display root for variable y |
| `RTN` | |
| `♦LBL 01` | Display out-of-limit message |
| `"ITER MAX REACHED"` | |
| `TONE 9` | |
| `PROMPT` | |
| `GTO 02` | |
| `♦LBL F` | Function f(x,y) |
| `X^2` | |
| `X<>Y` | |
| `3` | |
| `Y^X` | |

| *Program Step* | *Comment* |
|---|---|
| + | |
| 31 | |
| - | |
| RTN | |
| ♦LBL G | Function g(x,y) |
| * | |
| 6 | |
| - | |
| RTN | |